LAB 11

Objective: Explore and practice scheduling features.

This lab consists of three small experiments to highlight basic scheduling features.

Create a Pod using the **`redis.yaml`** manifest provided.The Pod will get scheduled automatically, the default scheduler does its job, and your Pod starts after the image gets downloaded.

Create a Pod using the **`redis-selector.yaml`** manifest provided. Note the **`nodeSelector`** field, which targets a node with the label **`foo=bar`**. You should see that the Pod remains in *pending* state. There are no nodes that match the **`nodeSelector`** labels.

Confirm this with:

```
$ kubectl describe pods foobar-node
```

How can you get this Pod scheduled?

Hint:

```
$ kubectl label node minikube foo=bar
```

Create a Pod using the **`redis-sched.yaml`** manifest provided. Note the **`schedulerName`** field in the manifest.

What happens? Why?

To get this Pod scheduled, you can run another scheduler. You can do this by starting the same container image like the default scheduler by setting the scheduler name option to **`foobar`**. For more details, please see this guide. You can also create a Pod Binding by hand.

Check the **`binding.json`** file in your lab directory. It basically defines a Node target **minikube** for the pod **`foobar-sched`**. By attaching this Binding to the Pod object, we will schedule the Pod by hand on **minikube**.

For simplicity, run a proxy in a separate terminal **`kubectl proxy --port=8080`**, then use **`curl`** to **`POST`** the Binding:

```
$ curl -H "Content-Type:application/json" -X POST --data @binding.json
http://localhost:8080/api/v1/namespaces/default/pods/foobar-sched/binding
/
```

Note the API endpoint of the Binding, referring to your Pod name: **`api/v1/namespaces/default/pods/foobar-sched/binding/`**.

